

Micro Musician

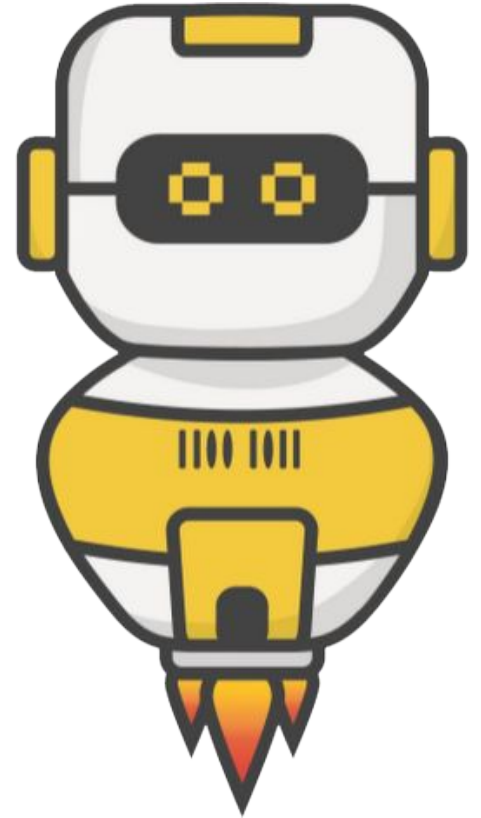
Mission 5



Pre-Mission Preparation

In the Mission 5 log, answer the pre-mission preparation questions:

- What are some ways a musician uses a computer to make music?
- How do you want to use music or sound in a CodeX program?

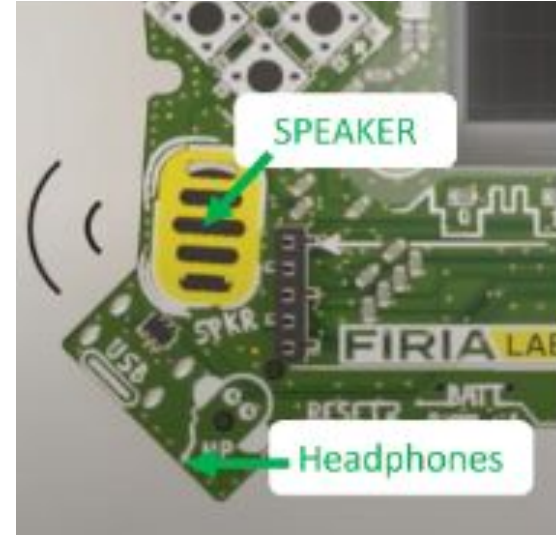


Objective #1: Sound outputs

There are two ways to listen to sound on the CodeX.

- Built-in speaker
- Plug in headphones

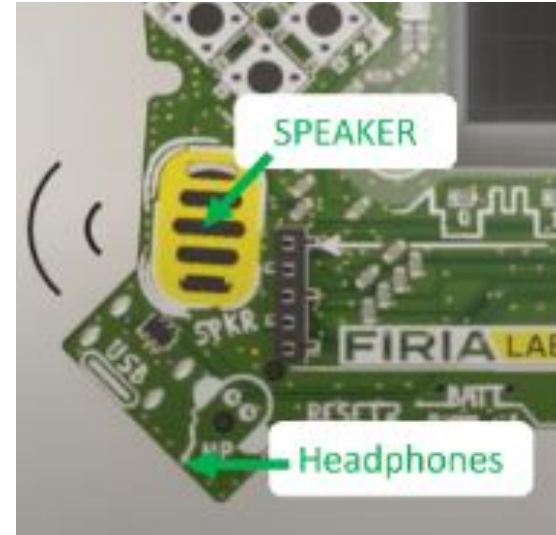
The CodeX uses a **codec chip** to change digital information into audio sound waves.



Objective #1: Sound outputs



With code you can:

- Play sound files
- Beep tones
- Control volume
- And more!



Mission Activity #1

DO THIS:

- Click on  to add it to your toolbox
- Scroll down in the toolbox until you find 
- Click on it and find the table with all the CodeX built-in sounds
- In your Mission Log, write down the names of sounds that you want to try

Example Audio Functions

The audio functions of the CodeX are exposed in the `audio` object from the `codex` module.

Try playing an MP3 file from the  `CodeX Sounds` collection:

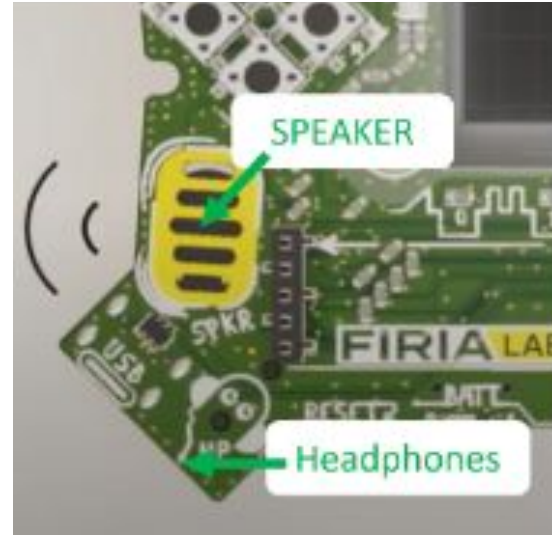


Mission Activity #1

Find the speaker and headphone jack.

DO THIS:

- Close the instruction panel
- Use camera controls to rotate the CodeX in the scene
- Click on the speaker
- Click on the headphone jack



Objective #2: Micro tunes

Now it is time to write code to play some sounds.

Start by playing an mp3 file.

- An mp3 is just an audio file in the mp3 format
- The CodeX has a few sample mp3 files already loaded

Here is an example:

```
from codex import *  
audio.mp3("sounds/welcome")
```



Mission Activity #2

DO THIS:

- Create a new file called **Music1**
- Type code to play an mp3 file
- You can use the one in the example, or pick a file from the sounds your are interested in trying
- Run the code

```
from codex import *  
audio.mp3("sounds/welcome")
```



Objective #3: Clean codes

Good code is easy to read by people.

As your programs get longer, you can do a few things to keep them readable:

- Use blank lines to separate sections of code
 - The computer ignores blank lines
- Add comments that explain what the code does
 - The computer ignores comments



Mission Activity #3

DO THIS:

- Add a blank line to your code
- Run the code

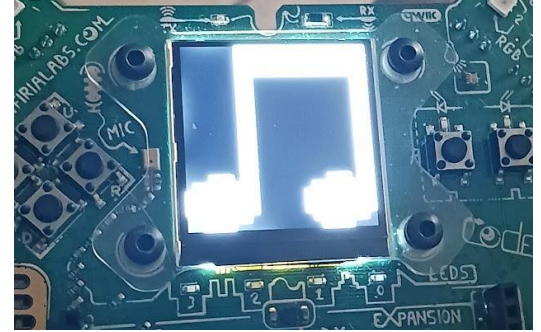
```
from codex import *  
  
audio.mp3("sounds/welcome")
```



Objective #4: Once more, with feeling

You don't want the screen to be blank when the sounds are playing.

- You can use **display.show()** with an image
- Use this statement BEFORE playing the mp3 file



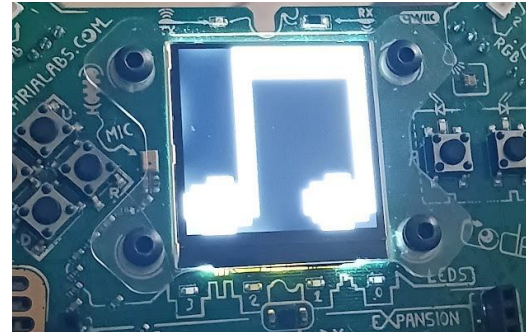
Mission Activity #4

Add an image to your code

DO THIS:

- Add `display.show(pics.MUSIC)`
- Change the audio file to “sounds/africa”
- Run your code
- Answer the question on the Mission Log

```
Music1 x
1 from codex import *
2
3 display.show(pics.MUSIC)
4 audio.mp3("sounds/africa")
5
```



Objective #5: Comments

Making code readable to people was mentioned in Objective 3.


You already learned about blank lines. Now let's find out about comments.


```
Music1 x
1 # This is what a comment looks like
2 from codex import *
3
4 # You can add a comment anywhere
5 # It explains what the code does
6 display.show(pics.MUSIC)
7 audio.mp3("sounds/africa")
8
9
```




Mission Activity #5

DO THIS:

- The instruction panel defines two words:
 - **Readability**
 - **Comments**
- Write the definitions in your Mission Log
- Click on  to add it to the toolbox

 **Readability**: Making code easy to understand for *humans*.

- Use descriptive variable names
- Use  **Comments** - notes in the code about what you're doing

In Python, anything that follows a `# to the end of the line`

... is a  **comment**, meaning it is *ignored* by the computer.



Mission Activity #5

Add three comments to your code

DO THIS:

- Add a comment with your name at the top of your code (line 1)
- Add a comment before the `display.show()` statement (line 4)
- Add a comment before the `audio.mp3()` statement (line 6)
- Run the code

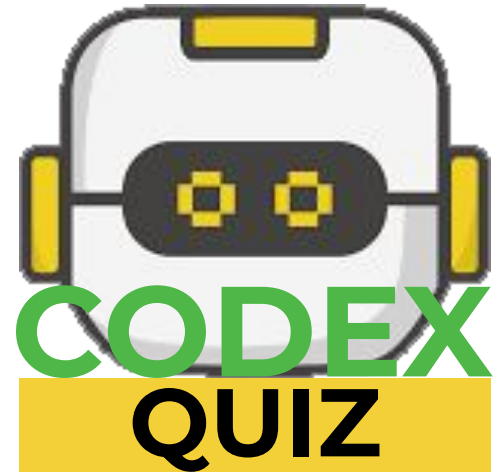
```
Music1 ×
1 # Student Name
2 from codex import *
3
4 # Display MUSIC pic
5 display.show(pics.MUSIC)
6 # Play Africa song
7 audio.mp3("sounds/africa")
8
```



Sounds and Readable quiz

During this mission you have learned about playing audio files and making your code readable.

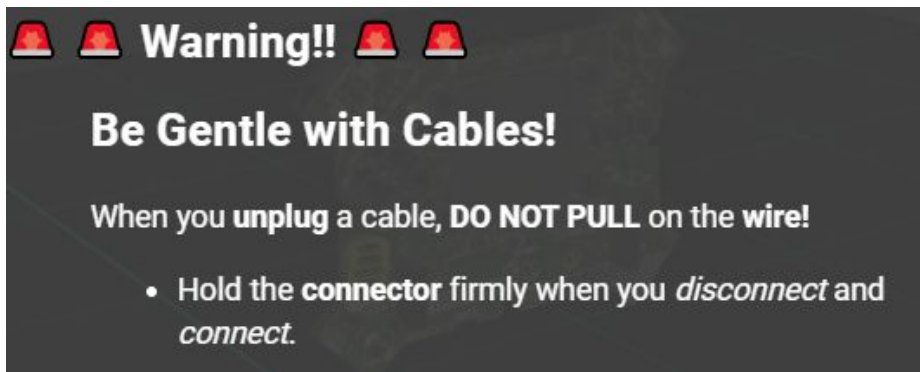
- Answer the 2 quiz questions



Objective #6: Portable mp3s

After the code is running on the CodeX, you can go unplugged.

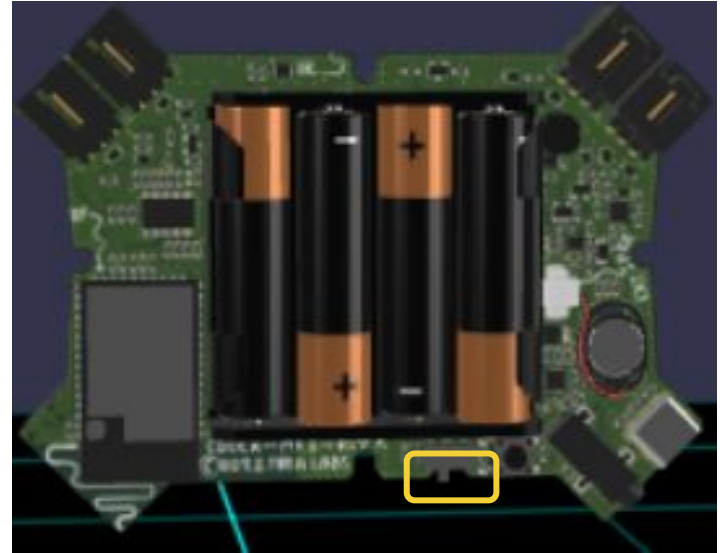
- After you run a program, it is loaded on the CodeX
- You can unplug the CodeX from the computer and run on batteries



Mission Activity #6

DO THIS:

- Close the instruction panel
- Use camera controls to rotate the CodeX in the scene
- Click on the BATT switch

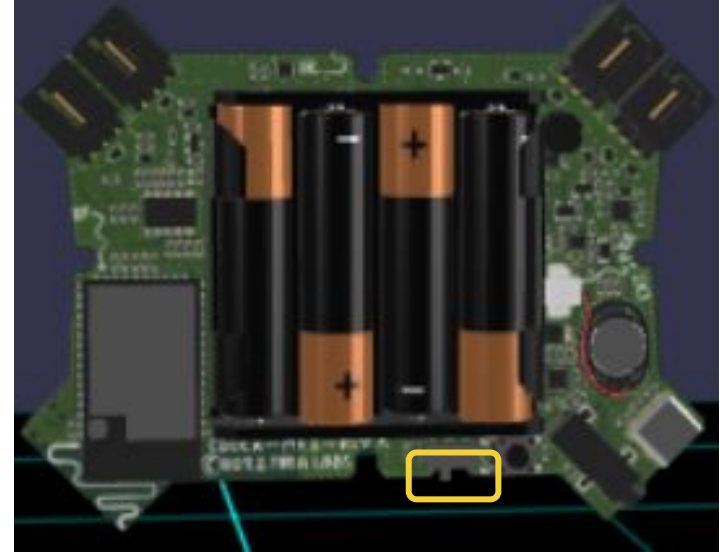


Mission Activity #6

(OPTIONAL)

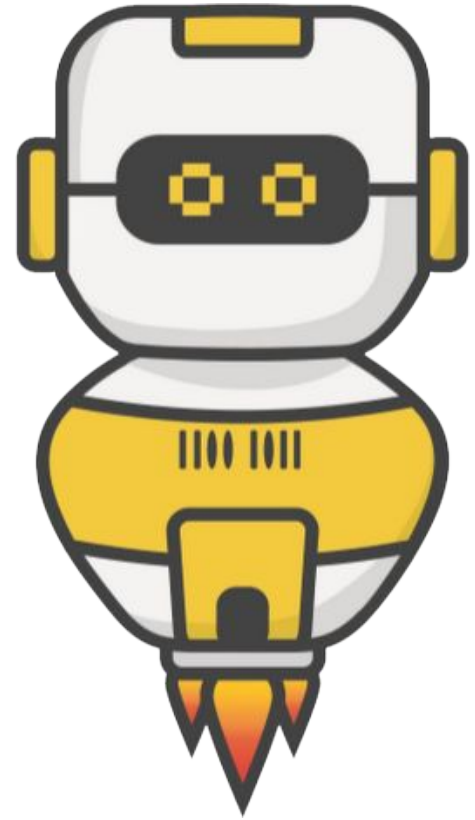
DO THIS:

- Put batteries in the CodeX
- Unplug from the computer
- Flip the BATT switch to position 1
- Enjoy your CodeX “unplugged”!



Post-Mission Reflection

- Read the “completed mission” message and click to complete the mission
- Complete the Mission 5 Log



Clearing your CodeX

Go to FILE -- BROWSE FILES
Select the “**Clear**” file and open it
Run the program to clear the CodeX



FIRIA LABS